**PHP基础**

**PHP数据类型**

1）PHP提供了8种常用的数据类型（四种基本数据类型，两种复合类型，两种特殊类型）

基本数据类型：interger、float、string、boolean

复合类型：array、object

特殊类型：resource、NULL

注：除对象类型外，其他类型的变量都可以不必显示地进行声明。

在PHP5中，默认情况下，对象是按引用传递的，其他类型的变量都是按值传递的。

2）数据类型转换分两种：1自动转换；2强制转换。强制转换有两种方式：1将需要的数据类型名写在变量前的括号内；2通过函数转换。

**常量与变量**

1）在PHP中，常量分自定义常量和预定义常量，而变量也可以分为自定义变量和预定义变量。对于预定义变量，在PHP4.2之后，可以利用超全局变量数组来访问预定义变量。PHP提供了9个可以直接使用的超全局变量数组，包括$\_SERVER、$\_GET、$\_POST、$\_COOKIE、$\_FILE、$\_ENV、$\_GLOBALS、$\_SESSION、$\_REQUEST。用户可以直接调用这些超全局变量数组来访问预定义变量

2）只有基本数据类型（interger、float、string、boolean）的值可以作为常量。

3）变量必须以$开头，常量必须不以$开头。

4）自定义常量使用define()函数定义。

5）变量是计算机内存中被命名的存储位置。

6）在PHP中通过变量给另一个变量赋值有两种方式：传值赋值和引用赋值。传值赋值仅将等号右侧变量的值赋给左侧变量；引用赋值将右侧变量的内存地址赋给左侧变量，即左右两侧的变量实际上指向的是同一个存储位置（左侧变量相当于右侧变量的一个别名），采用引用方式需要使用引用运算符&。

7）变量有作用域和生存期的概念，作用域是空间上的概念，生存期是时间上的概念。

8）在函数中定义的变量，作用域是该函数内部；在函数外定义的变量在该PHP文件中任何地方都有效。

在默认情况下不允许在函数内部直接访问全局变量，为了让一个函数可以访问全局变量，必须在该函数中显式地声明该全局变量，声明全局变量使用关键字global。

在函数中定义的变量具有局部作用域，在函数外定义的变量具有全局作用域。超全局变量具有超全局作用域，其作用范围包括函数内部和外部，即超全局变量在函数内部和外部都是可见的。

9）由浏览器提供的信息，在PHP中视为外部变量，外部变量主要通过HTML表单来获得。HTML表单具有向web服务器提交数据信息的交互功能。表单将数值传递到服务器的传输方式可以分为GET和POST两种。

**运算符**

任何一种编程语言中，运算符都具有优先级和结合性两种属性。

算术运算符：+-\*/

逻辑运算符：!、&&、||、and、or、not、xor

比较运算符：==、===、!=、<>、<、>、<=、>=

位运算符：&、|、~、^、<<、>>

赋值运算符：=

字符串运算符：.

执行运算符：“

加一、减一运算符：++、–

错误控制运算符：@

三元运算符：?:

数组运算符：[]

类运算符：instanceof

**流程控制**

1）在任何一种编程语言中，都有三种流程控制：顺序结构、选择结构、循环结构

2）选择结构：if、switch

3）循环结构：while、do…while、for、foreach

4）通常在无法确定循环的具体次数时，使用while循环语句；当确切知道循环次数时，一般采用for循环。

5）foreach用于遍历数组中的元素。由于foreach是专门为数组而设计的语句，如果将其应用于其他数据类型将会产生错误。

6）使用break语句可以结束while、for、foreach、do…while或者switch结构的执行，即从结构中跳出。如果有多层结构嵌套，还可以在break关键字后面加上数字，指明要跳出的结构层数。

7）continue语句用来跳过本次循环中剩余的代码，即直接跳回循环的开始位置（条件判断处）。如果条件为true则开始下一次循环，否则退出循环。

8）exit语句可以结束当前整个脚本的执行，函数die()是exit语句的别名。

**数组**

1）PHP允许一个数组包括不同数据类型的元素。

2）根据索引类型的不同，PHP数组可分为数字索引数组和关联数组。

3）与其他语言中的数组不同，PHP不需要创建数组时指定数组的大小，甚至不需要在使用数组前先行声明。

4）数组（无论是数字索引数组还是关联数组）初始化通常有两种方法：直接为数组元素赋值；使用array()函数。

5）在对数组进行初始化时，如果所需数据保存在另一个数组中，可以简单地使用运算符“=”直接对数组元素进行复制；如果数据保存在磁盘文件中，则可以从磁盘文件直接载入到数组中；如果数据保存在数据库中，那么还可以从数据库中直接载入数组，完成数组的初始化。

6）使用字符串作为索引的数组即称为关联数组。

**函数**

1）函数分内置函数和用户自定义函数。另外类也分内置类（核心内置或者扩展内置，C语言编写）和用户自定义类（PHP语言编写）

2）PHP的优势之一是定义了大量的函数，可以在程序中直接使用。几乎每个要实现的功能都可以在PHP内置或扩展函数库中找到相应的函数。包括数学函数、变量处理函数、字符串函数、数组函数、日期和时间函数、图形处理函数、目录管理函数、文件系统函数、电子邮件函数、mysql数据库函数、ftp文件传输函数等等。

3）匿名函数：匿名函数也叫闭包函数，允许临时创建一个没有指定名称的函数。最经常用作回调函数的参数，当然也有其它应用的情况。

**PHP中的面向对象编程技术**

1）面向对象程序设计（OOP），改变了以往的软件开发策略，使编程的注意力重新从应用逻辑回到了数据，而这种改变使得应用程序更接近于现实世界。

2）对象的抽象是类，类的具体化（也称实例化）就是对象，即类的实例化就是对象。对象为类的实例。

3）面向对象技术有三个基本概念：封装、继承、多态性。

4）继承关系可分为单重继承和多重继承。单重继承时一个子类只能有一个父类，而多重继承时一个子类可以有多个父类。在PHP中不支持多重继承，即在PHP中每个类都只能继承自一个父类，但是一个父类是允许有多个子类的。

5）多态性使得对象能够根据所得到的参数决定具体执行哪个成员函数，而对外却可以提供统一的接口，多态性分为以下两种：

（1）编译时多态性：通过函数重载，即提供多个同名的类方法，但各方法的参数类型和个数不同

（2）运行时多态性：通过函数重写，即子类继承父类，并在子类中重写父类中的方法

注：PHP不支持编译时多态性，即函数重载，所以在PHP中多态性指的就是运行时多态性，即函数重写。

6）类由属性和方法两部分组成。

类的属性即成员变量，在PHP中成员变量必须使用var来定义，但是如果类的属性指定了访问控制，则关键字var需省略

7）在完成类的定以后，可以通过关键字“new”来创建一个类的实例。

8）若要使用类中定义的属性和方法，只需使用运算符“->”即可。其中运算符“->”左侧为类对象名，右侧为属性或方法，需要注意的是，如果在类定义时需要访问类内部自身定义的属性或方法，则需要使用“this”指针，“this->”表示存取当前类内部定义的变量或函数。

9）PHP支持类的访问控制，可以控制属性和方法的可见性。常用的控制类型包括public、private、protected。需要注意的是，如果类的属性指定了访问控制，则关键字var需省略。

（1）public：表示该属性和方法是公有的，在类的内部或外部都可以访问，该选项是默认选项。

（2）private：表示指定的属性或方法只能在类的内部进行访问，不能由类实例化的对象直接调用，也不能由类的子类使用，即在类之外不允许读取或改变private属性的变量，也不能调用private属性的函数。

（3）protected：表明其指定的属性或方法只有在该类或其子类中使用。

10）作用域分辨运算符“::”：使用“::”运算符可以在没有创建任何类的实例的情况下访问类中的静态成员、方法和常量。

11）在面向对象编程中，如果希望在类刚刚实例化时，能够自动初始化某些变量或者触发执行某些函数，可以使用构造函数。构造函数是在一个类被实例化时自动调用的。构造函数可以接受参数，能够在创建对象时赋值给指定的变量，构造函数还可以调用类方法或其他函数，甚至可以调用其他的构造函数（如父类的构造函数）。构造函数不是必须的，但是由于其作用特殊，所以经常使用\_\_construct()。

12）与构造函数功能相反，可以定义一个名为\_\_destruct()的析构函数。析构函数会在类对象被销毁前调用，通常用来设定对象被销毁前需要完成的一些操作。需要注意的是析构函数不能带有任何参数。

13）脚本执行结束时，PHP会自动销毁内存中的所有对象。因此，对于一般的类对象不需要显示地定义析构函数。但是如果类对象在实例化时创建了不太容易销毁的数据（如数据存储在数据库中而不是内存中，就应当定义一个析构函数，在类对象销毁时使这些数据能够正常被销毁。）

14）有时，可能需要创建供所有类实例共享的属性和方法，这些属性和方法与所有的类实例有关，但不能设定仅由特定的类实例调用，这样的属性和方法在PHP中称为静态属性和方法。PHP中使用关键字“static”来定义静态属性和方法。静态属性和方法不能使用关键字this来引用，而需要使用关键字“self”。类中静态属性和方法使用“self::”来引用，而不再是“this->”，如果使用“this->”引用静态属性和方法则导致语法错误。

静态成员（方法及属性）属于类，而不属于任何类实例，但是所有类实例都能共享。通过类实例直接访问静态成员是不合常规的。

非静态方法可以访问静态属性，因为静态属性所有类实例都能共享；静态方法不可以访问非静态属性，因为静态方法属于类，而不是类实例。

对于何时定义为静态属性：当一个属性需要被所有类实例共享时。

对于何时定义为静态方法：当一个方法的程序体没有和这个类的非静态成员属性和非静态成员方法相关时就可以将其定义为静态方法，这样有利于提高程序的性能，并且也不必先new一个对象。

15）使用类运算符instanceof可以判断一个对象是否是一个类的实例。对象 instanceof类

对象 instanceof 对象所属类名及其父类名返回真，其它则为假。

16）魔术方法：\_\_construct, \_\_destruct, \_\_call, \_\_callStatic, \_\_get, \_\_set, \_\_isset, \_\_unset, \_\_sleep, \_\_wakeup, \_\_toString, \_\_set\_state 和 \_\_clone 等。

**PHP中的类继承**

1）继承允许在一个已有类的基础上建立新的类，并增加新的属性和方法。在PHP中使用关键字extends来实现类的继承。结构为：

class 子类名 extends 父类名

{

……

}

2）类的继承时单向的，即子类可以从父类继承特性，但父类却不能从子类继承特性。

3）函数重载是实现编译时的多态性，函数重写是在子类中重写父类中的函数。

4）类的重写（属性或方法的重写）：类的重写是指在子类中再次定义与父类相同的属性或方法。类的重写使得在子类中可以给某个属性赋予一个与其父类中不相同的值，也可以给某个方法指定一个与父类中不相同的功能。

5）一个子类可以通过继承得到父类中的属性和方法，并能够重写所继承的属性和方法。但在有些时候，在子类中会需要直接调用父类中的方法。在PHP中可以使用关键字parent来实现对父类中的属性和方法的访问。

6）当在一个成员函数定义前面使用关键字final时，表示该函数将不能被任何子类重写；同样地，如果不希望让一个类被继承，只需在定义类时为其添加final关键字。

7）PHP中提供了许多类函数可以帮助开发人员管理和使用类。经常使用的类函数包括：class\_exist()、get\_class()、get\_class\_vars()、get\_class\_methods()、get\_declared\_classes()、get\_object\_vars()、get\_parent\_class()、is\_a()、is\_subless\_of()和method\_exists()。

8）类常量不能有访问控制（public、protected、private），因为public、protected、private是用来控制类属性和方法的。

**PHP面向对象的高级特性**

1）PHP还具备一些更高级的OOP特征，包括抽象类与抽象方法、接口、对象的复制、自动加载类、序列化、反射等。这些高级OOP特征，使PHP对面向对象技术的支持更加完善。

2）抽象类是不能实例化的类，只能作为其他类的父类使用，其实质是对一些具有相似特征的类的泛化。抽象方法是抽象类中为其子类定义的一个空方法，在抽象类中并没有实现该方法，所有继承该抽象类的子类都必须包含该方法。子类可以实现该抽象方法，也可以继续留给其子孙实现。

3）在PHP中，使用关键字“abstract”定义一个抽象类和抽象方法。包含抽象方法的类必须定义为抽象类，但抽象类并不一定包含抽象方法。

4）接口（interface）是一种特殊的抽象类。这种抽象类中只包含抽象方法，不能定义属性（但可以定义常量），属性的定义和抽象方法的实现交给实现接口的类来完成，在接口中的抽象方法只能是public的，默认也是public权限。因为PHP只支持单向继承，一个类只能继承一个父类。为了解决这个问题，PHP引入了接口，一个类可以实现多个接口。将要实现的多个接口之间使用逗号分隔开，在子类（注意不是子接口）中要将所有接口中的抽象方法全部实现（不允许再遗留给实现接口的类的子类）才可以实例化对象，否则会报一个fatal错误。实现多个接口的格式如下：

class 类名 implements 接口一,接口二,接口三,……,接口n {

//实现所有接口中的抽象方法

}

实现多个接口是使用“implements”，同时还可以使用“extends”关键字继承一个类，格式如下：

class 类名 extends 父类名 implements 接口一,接口二,……,接口n {

//实现所有接口中的抽象方法

}

接口也可以继承，interface 接口2 extends 接口1{

//增加新的抽象方法

}

Implements（实现）、extends（继承）要分清二者之间的区别。

5）对象复制可以通过clone关键字来完成，当复制完成时，如果类中定义了\_\_clone()方法，则新创建的对象（复制生成的对象）中的\_\_clone()方法会被调用。

6）后期绑定：从PHP5.3.0开始，PHP增加了一个叫做后期绑定的功能。后期绑定的意思是说，static::不再被解析为定义当前方法所在的类，而是实际运行时计算的。有时也被称为静态绑定，因为它也可以用于静态方法的调用（但不仅限于静态方法）。

后期绑定最初是打算引入新的关键字来表示的，最后决定不引入新的关键字，而是用已有的static。

7）PHP中的句柄（handle）

this ：指向当前对象的句柄

self ：指向当前类的句柄

parent ：指向父类的句柄

另外在PHP后期绑定中，还有一个static句柄，static句柄的指向是运行时计算的，指向继承关系中的一个类的句柄。

**PHP5中的新特性**

1. 在PHP4中，对象默认是按值传递的；在PHP5中，对象默认是按引用传递的

2）在PHP5中，只有对象类型和数组类型，可以在定义函数的时候在形参中添加类型约束。注意，即使使用了类型约束，如果使用NULL作为参数的默认值，那么在调用函数的时候依然可以使用NULL作为实参。